
ZINGA
MAKE IT EASY!

0.

ACKNOWLEDGEMENT

I undertook this project work, as a part of my class 12th computer science practicals. I would like to express my gratitude and thanks to my computer science teacher Mrs.Ananthy and my parents DharmaRao.G and Lakshmi.G for their continuous support throughout this project. And a special thanks to my sister Menaja G and my uncle Kusa Raju. G for motivating me for my first project development and, also to all those who contributed directly or indirectly towards the completion of my school project.

1.

ABSTRACT

The main aim of creating this project is to collect the test details of the students in a class and to display, delete, select, update, and reset the records from the student table, it doesn't require any paperwork to be done, and the teacher can access the test details for the updation or deletion of marks of a student of a particular class at any time anywhere. for accessing a particular class scores teachers can login with their respective ID and passwords given.

(ID- ananthy , Password- 1234)

Understanding 'ZINGA' the application program :-

- It is user-friendly, as it works based on GUI (Graphical User Interface) using Tkinter
- It can store as many records as possible and generate them whenever required.
- It can perform all operations required like resetting, the addition of data, displaying data, updating data, deleting data, searching data and etc.., which saves time.

2.

REQUIREMENT ANALYSIS

• HARDWARE REQUIREMENTS

- **Processor** 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz 2.42 GHz
- **Installed RAM** 16.0 GB (15.8 GB usable)
- **Operating system** Windows 11

• SOFTWARE REQUIREMENTS

- **System type** 64-bit operating system, x64-based processor
- **Software used** Python version 3.9**,** MySQL 8.0, Word document

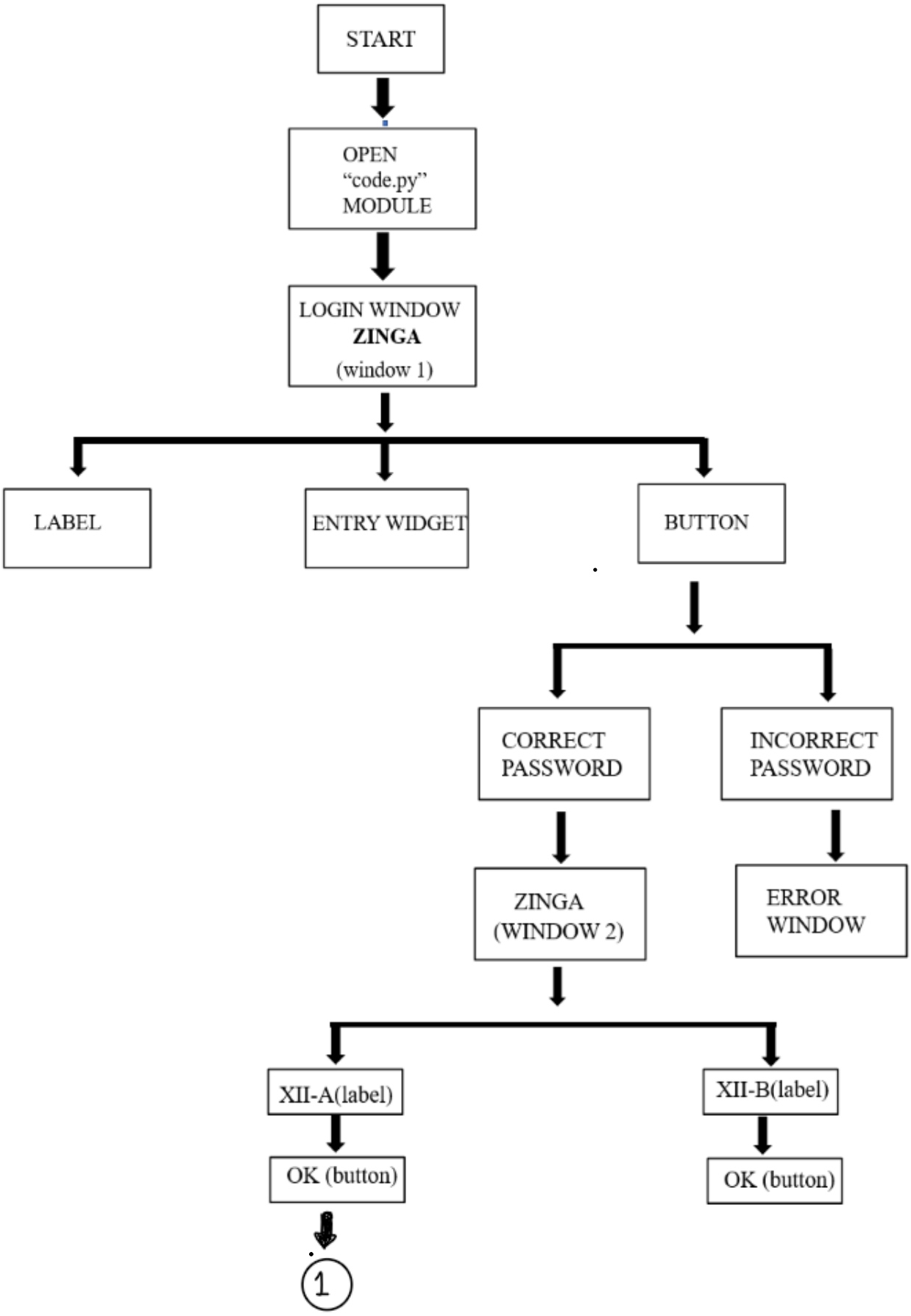
3.

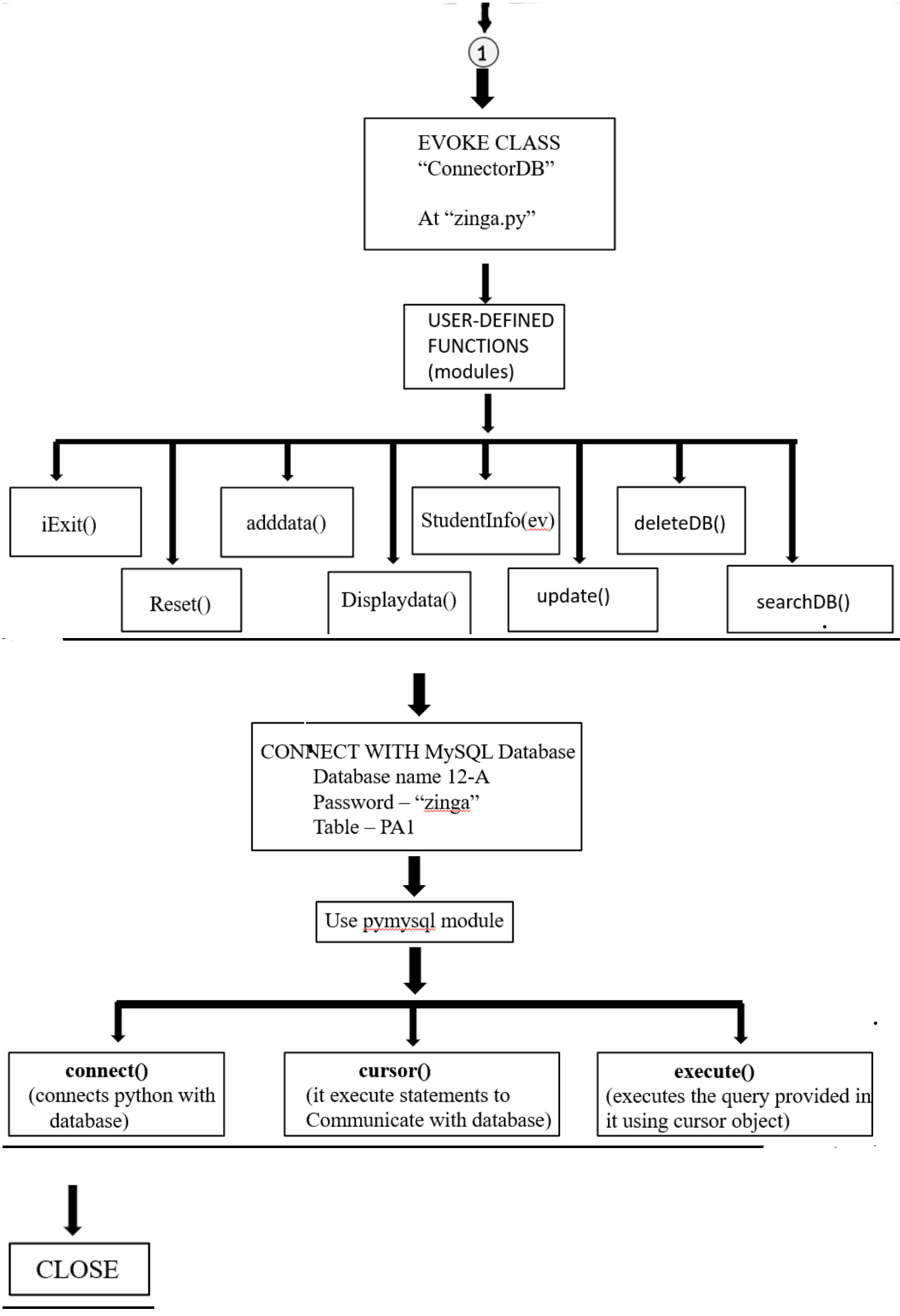
APPLICATION

As this application is used to collect the test scores of cyclic tests of all the students in a class. we can use it in different Schools, colleges, and many institutions. Here we use front as python, which can be used by all the teachers as it is GUI friendly which allows all teachers to comfortably work by using the front end at python and the back end as MySQL database which requires no need of any Querying or programming knowledge to inserts various details into the ‘PA 1’ table.

4.DESIGN

1) BLOCK DIAGRAM




1
EVOKE CLASS
"ConnectorDB"
At "zinga.py"
USER-DEFINED
FUNCTIONS
(modules)
iExit()
Reset()
adddata()
Displaydata()
StudentInfo(ev)
update()
deleteDB()
searchDB()
CONNECT WITH MySQL Database
Database name 12-A
Password – "zinga"
Table – PA1
Use pymysql module
connect()
(connects python with database)
cursor()
(it execute statements to Communicate with database)
execute()
(executes the query provided in it using cursor object)
CLOSE

2)EXPLANATION

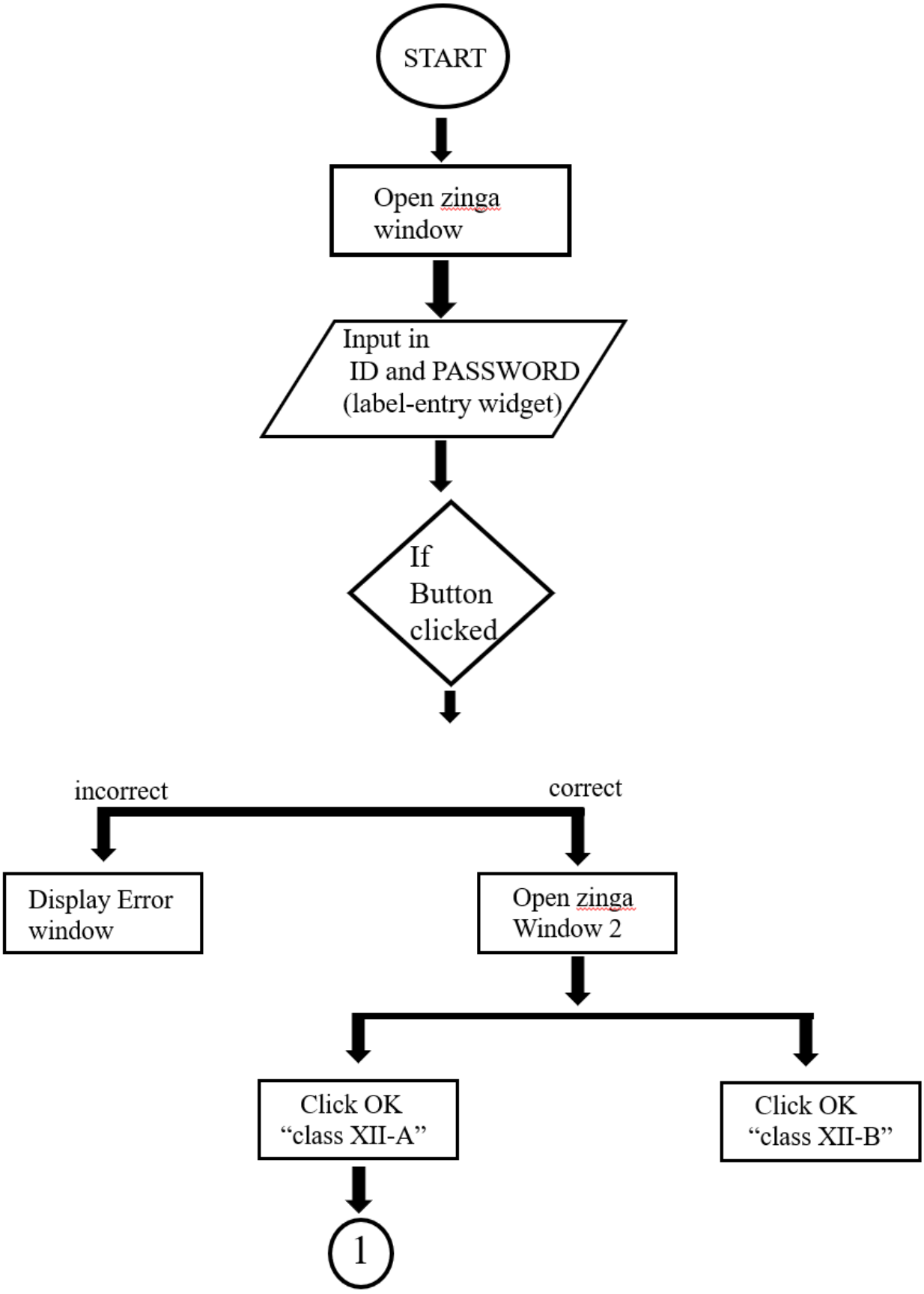
We give the inputs from the software(zinga) through the front end. We can either add, update, clear, reset or delete a record from the table, we can store it in the memory of the system MySQL database which can store a large amount of data, which makes teachers or the staff work easily without requirement of any technical knowledge as everything is front end which makes them easy to update or insert the details of the student scores easily.

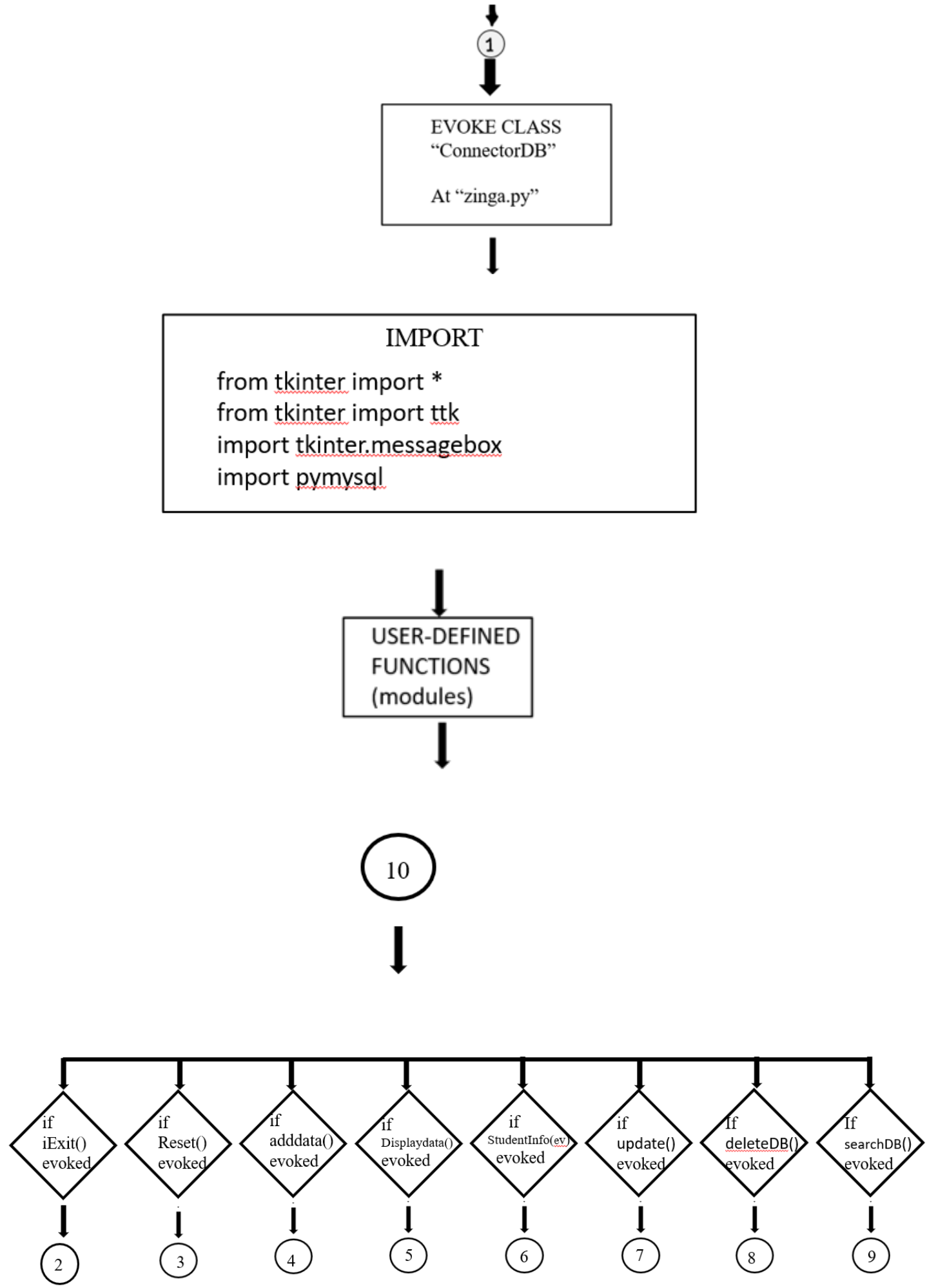
Here in order to work successfully with this application we use two user-defined modules namely code.py and zinga.py

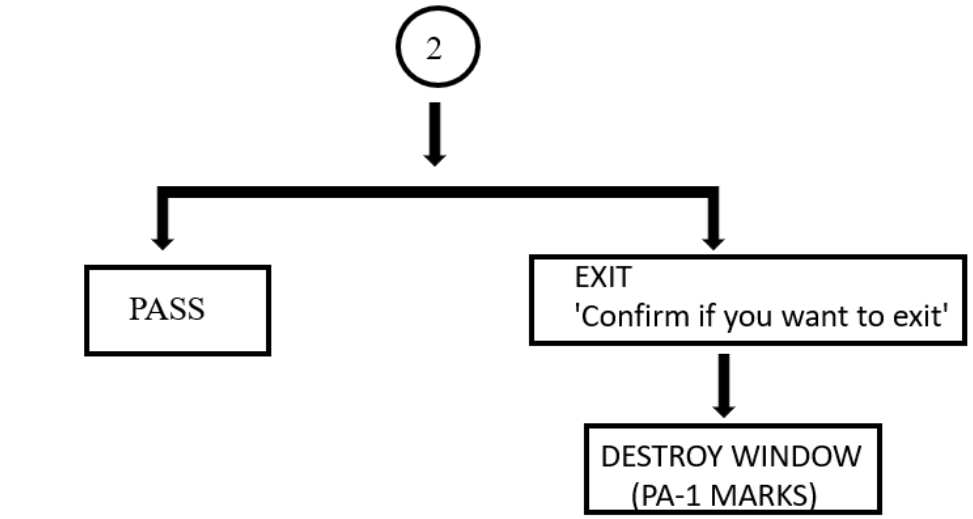
In the first module “code”, we use it for creating the login window by importing tkinter where we place various widgets such as labels, buttons, entry widgets and etc… and on clicking the button opens another window that asks for which class mark to be chosen and further of the program flow which evokes the “ConnectorDB” which is a class been defined at the 2nd module “zinga.py” which contains a lot of functions in order to insert, update ,reset,or delete the record.

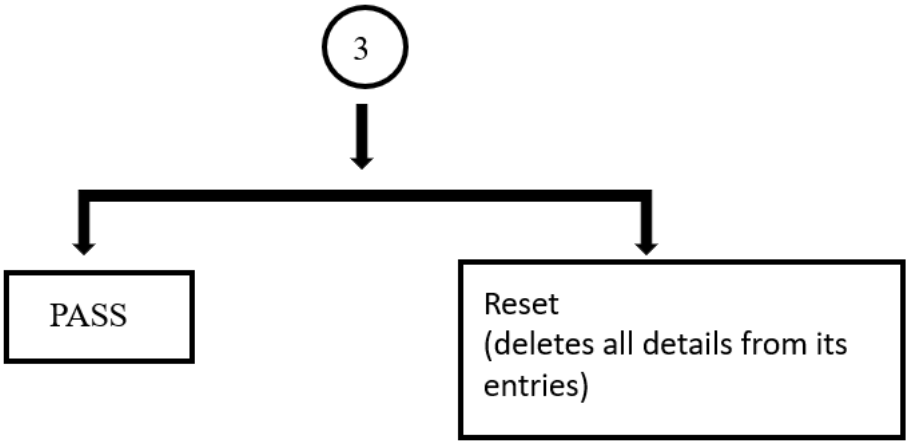
5.FLOWCHART

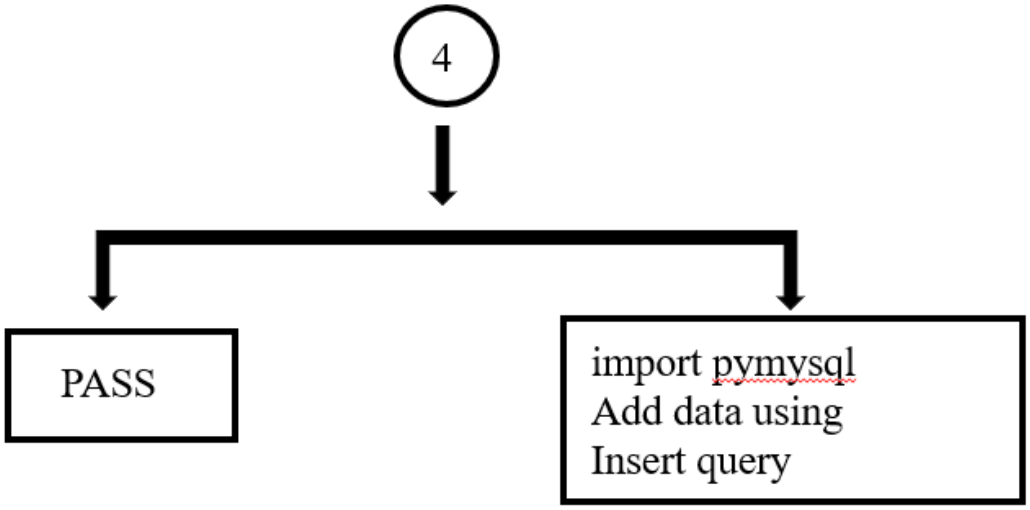
❖ **PROJECT VIEW**

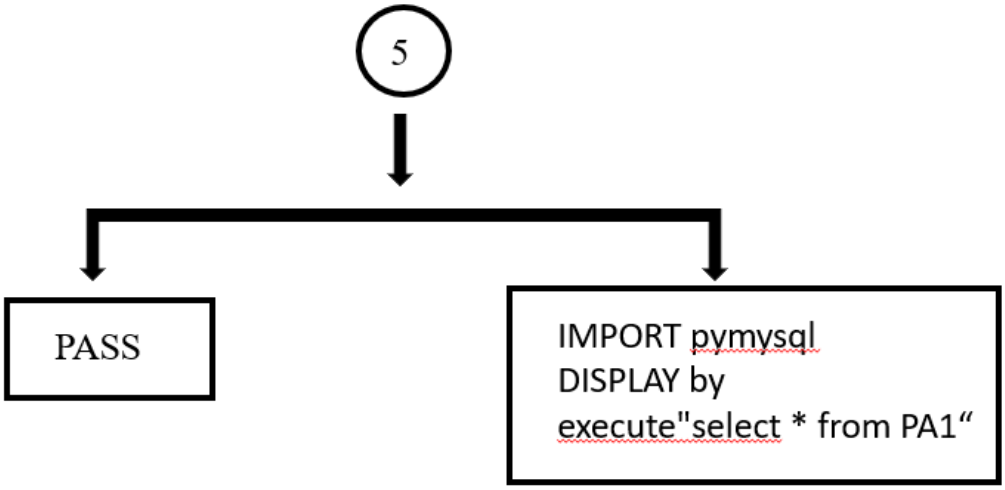


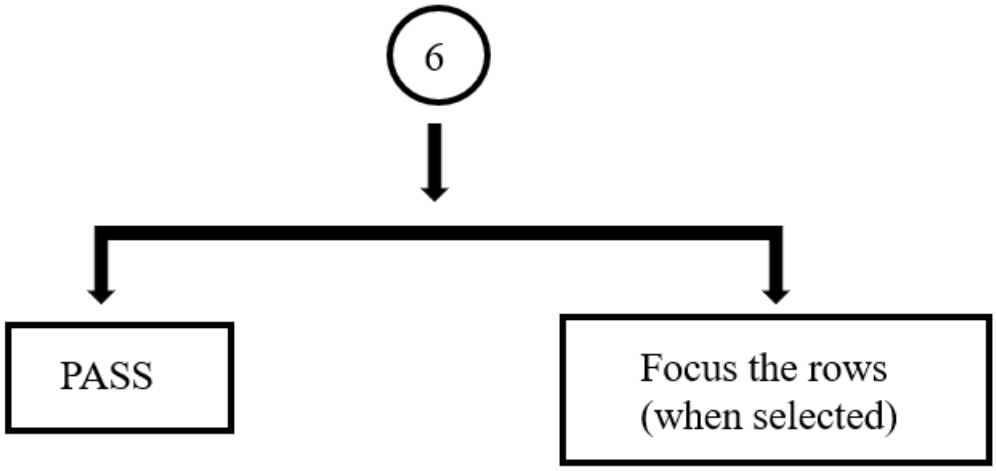

1
EVOKE CLASS
"ConnectorDB"
At "zinga.py"
IMPORT
from tkinter import *
from tkinter import ttk
import tkinter.messagebox
import pymysql
USER-DEFINED
FUNCTIONS
(modules)
10
if
iExit()
evoked
if
Reset()
evoked
if
adddata()
evoked
if
Displaydata()
evoked
if
StudentInfo(ev)
evoked
if
update()
evoked
If
deleteDB()
evoked
If
searchDB()
evoked
2
3
4
5
6
7
8
9

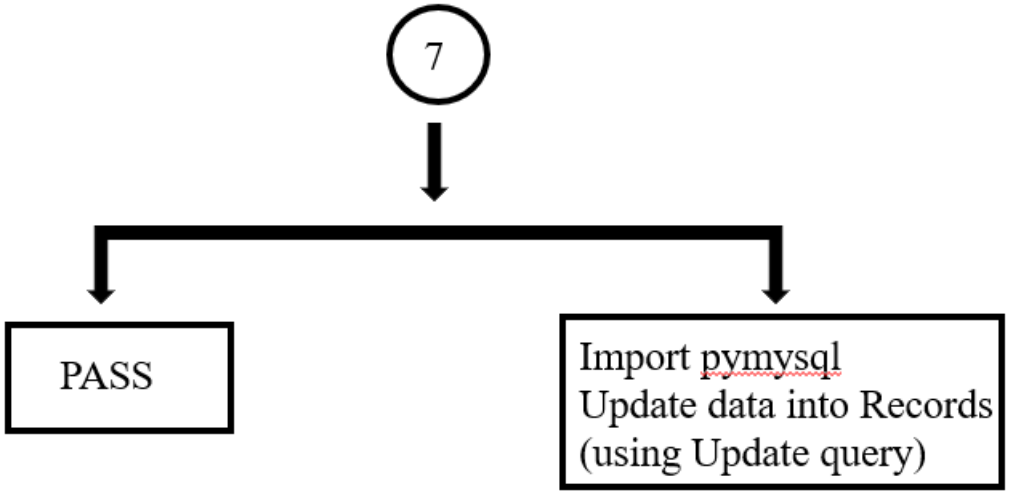

2
PASS
EXIT
'Confirm if you want to exit'
DESTROY WINDOW
(PA-1 MARKS)

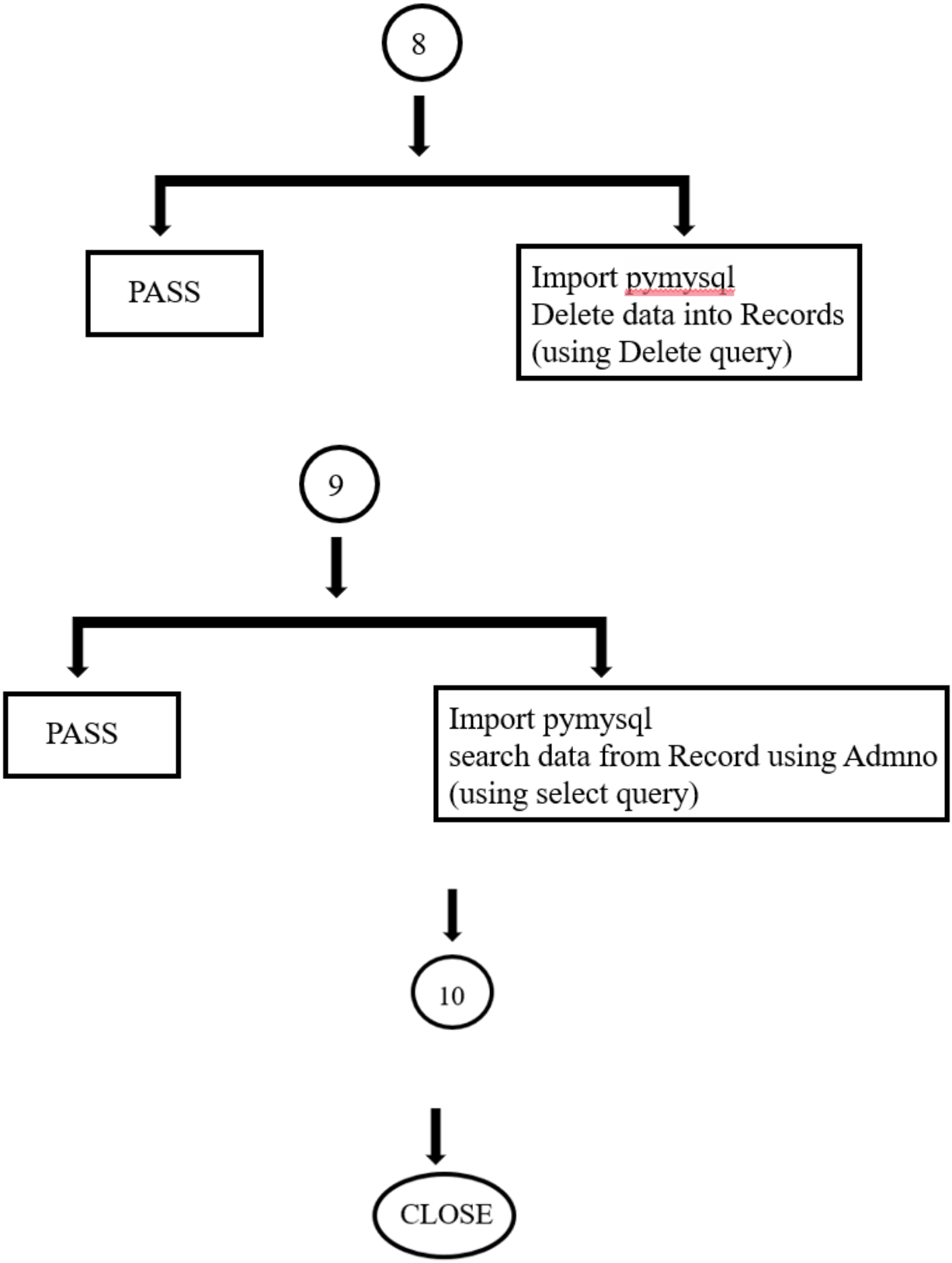

3
PASS
Reset
(deletes all details from its entries)


4
PASS
import pymysql
Add data using
Insert query


5
PASS
IMPORT pymysql
DISPLAY by
execute"select * from PA1“


6
PASS
Focus the rows
(when selected)


7
PASS
Import pymysql
Update data into Records
(using Update query)


8
PASS
Import pymysql
Delete data into Records
(using Delete query)
9
PASS
Import pymysql
search data from Record using Admno
(using select query)
10
CLOSE

6.ALGORITHM

Step 1: Start

Step 2: Open module “code”

Step 3: import tkinter

Step 4: import tkinter.messagebox

Step 5: import pymysql

Step 6: import tkinter ttk

Step 7: Create window

Step 8: Create Label, Entry widget

Step 9: Create button and give command

Step 10: Open module “Zinga”

Step 11: Create ‘class ConnectDB’

Step 12: Create top frame, bottom frame ,left frame, right frame

Step 13: Create some user-defined functions like ‘selectDb’,’update’, ‘deleteDB’, ‘Exit’, ‘Reset’

Step 14: evoking “class ConnectDB”

Step 15: Stop

7.

SOURCE CODE

At module code.py

```
from tkinter import *
from tkinter import ttk
import tkinter.messagebox
import pymysql

#LOGIN WINDOW
login=tkinter.Tk()
login.title('ZINGA')
login.geometry('1900x1000')
login.configure(bg='blue')
#LABEL
l=tkinter.Label(login,text="LOGIN",font=('times new roman', 30),width=15,bg='white')
l.place(x=590,y=200)
l1=tkinter.Label(login,text="ID",font=('Helvetica bold', 15),width=12)
l1.place(x=550,y=350)
#ENTRY WIDGET
e=tkinter.Entry(login,width=23,font=(300))
e.place(x=700,y=350)
l2=tkinter.Label(login,text="PASSWORD",font=('Helvetica bold', 15),width=12)
l2.place(x=550,y=400)
e1=tkinter.Entry(login,width=23,font=(300),show='*')
e1.place(x=700,y=400)
login3=1
def create_login3():
    login2.destroy()
    from zinga import ConnectorDB
```

```
    if __name__=='__main__':
        root=Tk()
        application=ConnectorDB(root)
        root.mainloop()
login2=0
def create_login2():
    global login2
    login2=tkinter.Tk()
    login2.title('ZINGA')
    login2.geometry('1900x1000')
    login2.configure(bg='blue')
    l4=tkinter.Label(login2,text="MARKS",font=('Helvetica bold', 40,'bold'),width=20,bg='white')
    l4.place(x=500,y=200)
    l5=tkinter.Label(login2,text="XII-A",font=('Helvetica bold', 20),width=12,bg='white')
    l5.place(x=550,y=350)
    b2=tkinter.Button(login2,text='OPEN',font=('Helvetica bold', 20,'bold'),width=5,bg='white',
     command=create_login3)
    b2.place(x=950,y=340)
    l6=tkinter.Label(login2,text="XII-B",font=('Helvetica bold', 20),width=12,bg='white')
    l6.place(x=550,y=460)
    b3=tkinter.Button(login2,text='OPEN',font=('Helvetica bold', 20,'bold'),width=5,bg='white')
    b3.place(x=950,y=450)


def check():
    if e.get()=='ananthy' and e1.get()=='1234':
        login.destroy()
        create_login2()

    else:
        login1=tkinter.Tk()
        login1.geometry('300x200')
        login1.configure(bg='red')
        login1.title('login_failed')
        l3=tkinter.Label(login1,text='LOGIN FAILED !',font=('Helvetica bold', 20),bg='white')
        l3.place(x=30,y=100)
```

```
#BUTTON
b=tkinter.Button(login,text='OK',font=('Helvetica_bold',
20,'bold'),width=5,bg='white',command=check)
b.place(x=720,y=500)
```

At module zinga.py

```
from tkinter import *
from tkinter import ttk
import tkinter.messagebox
import pymysql


class ConnectorDB:
    def __init__(self,root):
        self.root=root
        titlespace=" "
        self.root.title(102 * titlespace +'PA-I Marks')
        self.root.geometry('1180x750')
        self.root.resizable(width=False,height=False)

        MainFrame=Frame(self.root,bd=10,width=770,height=700,relief=RIDGE,bg='blue')
        MainFrame.grid()

        TitleFrame=Frame(MainFrame,bd=7,width=770,height=100,relief=RIDGE)
        TitleFrame.grid(row=0,column=0)
        TopFrame3=Frame(MainFrame,bd=5,width=770,height=500,relief=RIDGE)
        TopFrame3.grid(row=1,column=0)

        LeftFrame=Frame(TopFrame3,bd=5,width=770,height=400,padx=2,
            bg='blue',relief=RIDGE)
        LeftFrame.pack(side=LEFT)
```

```
    LeftFrame1=Frame(LeftFrame,bd=5,width=600,height=180,padx=2,pady=4,
          relief=RIDGE)
    LeftFrame1.pack(side=TOP,padx=0,pady=0)

    RightFrame1=Frame(TopFrame3,bd=5,width=100,height=400,padx=2,
          bg='blue',relief=RIDGE)
    RightFrame1.pack(side=RIGHT)
    RightFrame1a=Frame(RightFrame1,bd=5,width=90,height=300,padx=2,
          pady=2,relief=RIDGE)
    RightFrame1a.pack(side=TOP)
#_____________________________________________________________________#

    Admno=StringVar()
    Rollno=StringVar()
    Name=StringVar()
    Mathematics=StringVar()
    Physics=StringVar()
    Chemistry=StringVar()
    Computer=StringVar()
    English=StringVar()



#_____________________________________________________________________#
    def iExit():
        iExit=tkinter.messagebox.askyesno('Pa-1 Marks','Confirm if you want to exit')
        if iExit>0:
            root.destroy()
            return
    def Reset():
        self.entAdmno.delete(0,END)
        self.entRollno.delete(0,END)
        self.entName.delete(0,END)
        self.entMathematics.delete(0,END)
        self.entPhysics.delete(0,END)
```

```
    self.entChemistry.delete(0,END)
    self.entComputer.delete(0,END)
    self.entEnglish.delete(0,END)
def adddata():
    if Admno.get()=="" or Rollno.get()=="" or Name.get()=="" or
        Mathematics.get()=="" or_Physics.get()=="" or Chemistry.get()=="" or
        Computer.get()=="" or English.get()=="":
      tkinter.messagebox.showerror('Pa-1 Marks','Enter Correct Details')
    else:
      import pymysql
      sqlcon= pymysql.connect(host='localhost',user='root',passwd='zinga',
              database='12A')
      cur = sqlcon.cursor()
      cur.execute("insert_into_PA1_values_(%s,%s,%s,%s,%s,%s,%s,%s)",
                (Admno.get(),Rollno.get(),Name.get(),Mathematics.get(),
                 Physics.get(),Chemistry.get(),Computer.get(),English.get()))
      sqlcon.commit()
      sqlcon.close()
      tkinter.messagebox.showinfo('Pa-1 Marks','Record Entered Successfully')

def displaydata():
    sqlcon= pymysql.connect(host='localhost',user='root',passwd='zinga',
            database='12A')
    cur = sqlcon.cursor()
    cur.execute("select * from PA1")
    result=cur.fetchall()
    if len(result)!=0:
      self.student_records.delete(*self.student_records.get_children())
      for row in result:
        self.student_records.insert('',END,values=row)
      sqlcon.commit()
    sqlcon.close()
    #tkinter.messagebox.showinfo('Pa-1 Marks','Record Entered Successfully')
def StudentInfo(ev):
```

```
    viewInfo= self.student_records.focus()
    learnerData= self.student_records.item(viewInfo)
    row=learnerData['values']
    Admno.set(row[0])
    Rollno.set(row[1])
    Name.set(row[2])
    Mathematics.set(row[3])
    Physics.set(row[4])
    Chemistry.set(row[5])
    Computer.set(row[6])
    English.set(row[7])
def update():
    sqlcon= pymysql.connect(host='localhost',user='root',passwd='zinga',
                database='12A')
    cur = sqlcon.cursor()
    cur.execute("update PA1set rollno=%s,Name=%s,Mathematics=%s,Physics=%s,
                    Chemistry=%s,Computer=%s, English=%s where admno=%s",
                    (Rollno.get(),Name.get(),Mathematics.get(),Physics.get(),
                     Chemistry.get(),Computer.get(),English.get(),Admno.get()))
    sqlcon.commit()
    sqlcon.close()
    tkinter.messagebox.showinfo('data entry form','Record updated succesfully')

def deleteDB():
    sqlcon= pymysql.connect(host='localhost',user='root',passwd='zinga',
                database='12A')
    cur = sqlcon.cursor()
    cur.execute("delete from pa1 where admno=%s",Admno.get())
    sqlcon.commit()
    displaydata()
    sqlcon.close()
    tkinter.messagebox.showinfo('data entry form','Record Deleted succesfully')
    Reset()
```

```
    def searchDB():
        try:
            sqlcon= pymysql.connect(host='localhost',user='root',passwd='zinga',
                    database='12A')
            cur = sqlcon.cursor()
            cur.execute("select * from pa1 where admno=%s",Admno.get())
            row=cur.fetchall
            Admno.set(row[0])
            Rollno.set(row[1])
            Name.set(row[2])
            Mathematics.set(row[3])
            Physics.set(row[4])
            Chemistry.set(row[5])
            Computer.set(row[6])
            English.set(row[7])

            sqlcon.commit()
        except:
            tkinter.messagebox.showinfo('data entry form','No such record Found')
            Reset()
        sqlcon.close()



#_____________________________________________________________________#
    self.lbltitle=Label(TitleFrame,font=('arial',40,'bold'),text='PA-I',bd=7)
    self.lbltitle.grid(row=0,column=0,padx=132)

#_____________________________________________________________________#
```

```
self.lblAdmno=Label(LeftFrame1,font=('arial',15,'bold'),text='Admno',bd=7)
self.lblAdmno.grid(row=0,column=0,sticky='',padx=132)
self.entAdmno=Entry(LeftFrame1,font=('arial',15,'bold'),bd=5,width=44,
                    justify='left', textvariable=Admno)
self.entAdmno.grid(row=0,column=1,sticky='',padx=5)

self.lblRollno=Label(LeftFrame1,font=('arial',15,'bold'),text='Rollno',bd=7)
self.lblRollno.grid(row=1,column=0,sticky='',padx=132)
self.entRollno=Entry(LeftFrame1,font=('arial',15,'bold'),bd=5,width=44,justify='left',
                 textvariable=Rollno)
self.entRollno.grid(row=1,column=1,sticky='',padx=5)


self.lblName=Label(LeftFrame1,font=('arial',15,'bold'),text='Name',bd=7)
self.lblName.grid(row=2,column=0,sticky='',padx=132)
self.entName=Entry(LeftFrame1,font=('arial',15,'bold'),bd=5,width=44,justify='left',
                textvariable=Name)
self.entName.grid(row=2,column=1,sticky='',padx=5)

self.lblMathematics=Label(LeftFrame1,font=('arial',15,'bold'),
                          text='Mathematics',bd=7)

self.lblMathematics.grid(row=3,column=0,sticky='',padx=132)
self.entMathematics=Entry(LeftFrame1,font=('arial',15,'bold'),bd=5,
                          width=44,justify='left', textvariable=Mathematics)
self.entMathematics.grid(row=3,column=1,sticky='',padx=5)

self.lblPhysics=Label(LeftFrame1,font=('arial',15,'bold'),text='Physics',bd=7)
self.lblPhysics.grid(row=4,column=0,sticky='',padx=132)
self.entPhysics=Entry(LeftFrame1,font=('arial',15,'bold'),bd=5,
                    width=44,justify='left', textvariable=Physics)
self.entPhysics.grid(row=4,column=1,sticky='',padx=5)

self.lblChemistry=Label(LeftFrame1,font=('arial',15,'bold'),text='Chemistry',bd=7)
```

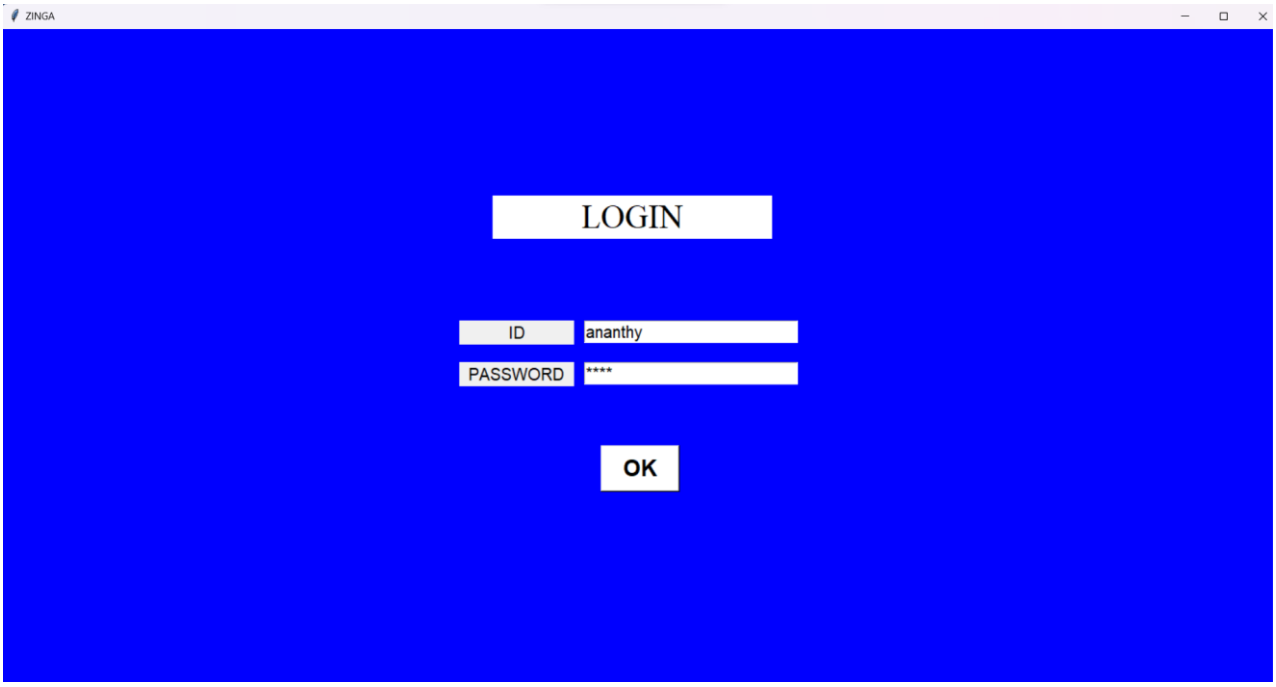
```
self.lblChemistry.grid(row=5,column=0,sticky='',padx=132)
self.entChemistry=Entry(LeftFrame1,font=('arial',15,'bold'),bd=5,
                        width=44,justify='left',textvariable=Chemistry)
self.entChemistry.grid(row=5,column=1,sticky='',padx=5)

self.lblComputer=Label(LeftFrame1,font=('arial',15,'bold'),text='Computer',bd=7)
self.lblComputer.grid(row=6,column=0,sticky='',padx=132)
self.entComputer=Entry(LeftFrame1,font=('arial',15,'bold'),bd=5,
                        width=44,justify='left',textvariable=Computer)
self.entComputer.grid(row=6,column=1,sticky='',padx=5)

self.lblEnglish=Label(LeftFrame1,font=('arial',15,'bold'),text='English',bd=7)
self.lblEnglish.grid(row=7,column=0,sticky='',padx=132)
self.entEnglish=Entry(LeftFrame1,font=('arial',15,'bold'),bd=5,
                      width=44,justify='left', textvariable=English)
self.entEnglish.grid(row=7,column=1,sticky=W,padx=5)#W
#CREATING BUTTONS
#_____TABLE TREEVIEW_____@ttk module_____#
scroll_y=Scrollbar(LeftFrame,orient=VERTICAL)

self.student_records=ttk.Treeview(LeftFrame,height=12,
                           columns=('Admno','Rollno','Name','Mathematics','Physics',
                          'Chemistry','Computer','English' ),
                          yscrollcommand=scroll_y.set)
scroll_y.pack(side=RIGHT,fill=Y)#Y
self.student_records.heading('Admno',text='Admno')
self.student_records.heading('Rollno',text='Rollno')
self.student_records.heading('Name',text='Name')
self.student_records.heading('Mathematics',text='Mathematics')
self.student_records.heading('Physics',text='Physics')
self.student_records.heading('Chemistry',text='Chemistry')
self.student_records.heading('Computer',text='Computer')
self.student_records.heading('English',text='English')
```

```
        self.student_records['show']='headings'

        self.student_records.column('Admno',width=70)
        self.student_records.column('Rollno',width=100)
        self.student_records.column('Name',width=100)
        self.student_records.column('Mathematics',width=70)
        self.student_records.column('Physics',width=70)
        self.student_records.column('Chemistry',width=70)
        self.student_records.column('Computer',width=70)
        self.student_records.column('English',width=70)

        self.student_records.pack(fill=BOTH,expand=1)
        self.student_records.bind("<ButtonRelease-1>",StudentInfo)
        #displaydata()



#_____________________________BUTTONS_____________________________#
        self.btnADDNEW=Button(RightFrame1a,font=('arial',17,'bold'),
                              text='ADD NEW',bd=4,pady=1,padx=24,width=8,height=2,
                              command=adddata).grid(row=0,column=0,padx=1)
         self.btnDISPLAY=Button(RightFrame1a,font=('arial',17,'bold'),
                               text='DISPLAY',bd=4,pady=1, padx=24,width=8,height=2,
                               command=displaydata).grid(row=1,column=0,padx=1)
        self.btnUPDATE=Button(RightFrame1a,font=('arial',17,'bold'),
                              text='UPDATE',bd=4,pady=1, padx=24,width=8,height=2,
                              command=update).grid(row=2,column=0,padx=1)
        self.btnDELETE=Button(RightFrame1a,font=('arial',17,'bold'),
                              text='DELETE',bd=4,pady=1, padx=24,width=8,height=2,
                              command=deleteDB).grid(row=3,column=0,padx=1)
        self.btnSEARCH=Button(RightFrame1a,font=('arial',17,'bold'),
                              text='SEARCH',bd=4,pady=1, padx=24,width=8,height=2,
                              command=searchDB).grid(row=4,column=0,padx=1)
```

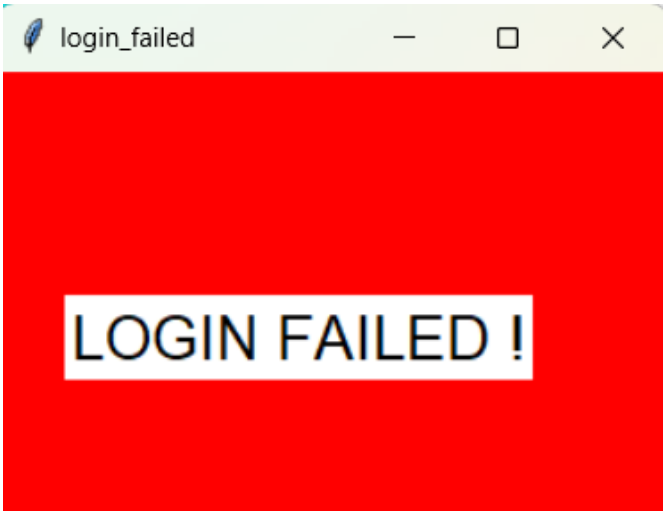
```
self.btnRESET=Button(RightFrame1a,font=('arial',17,'bold'),
                     text='RESET',bd=4,pady=1, padx=24,width=8,height=2,
                     command=Reset).grid(row=5,column=0,padx=1)
self.btnEXIT=Button(RightFrame1a,font=('arial',17,'bold'),
                    text='EXIT',bd=4,pady=1,padx=24,width=8,height=2,
                    command=iExit).grid(row=6,column=0,padx=1)
```

8.SAMPLE SCREENSHOTS

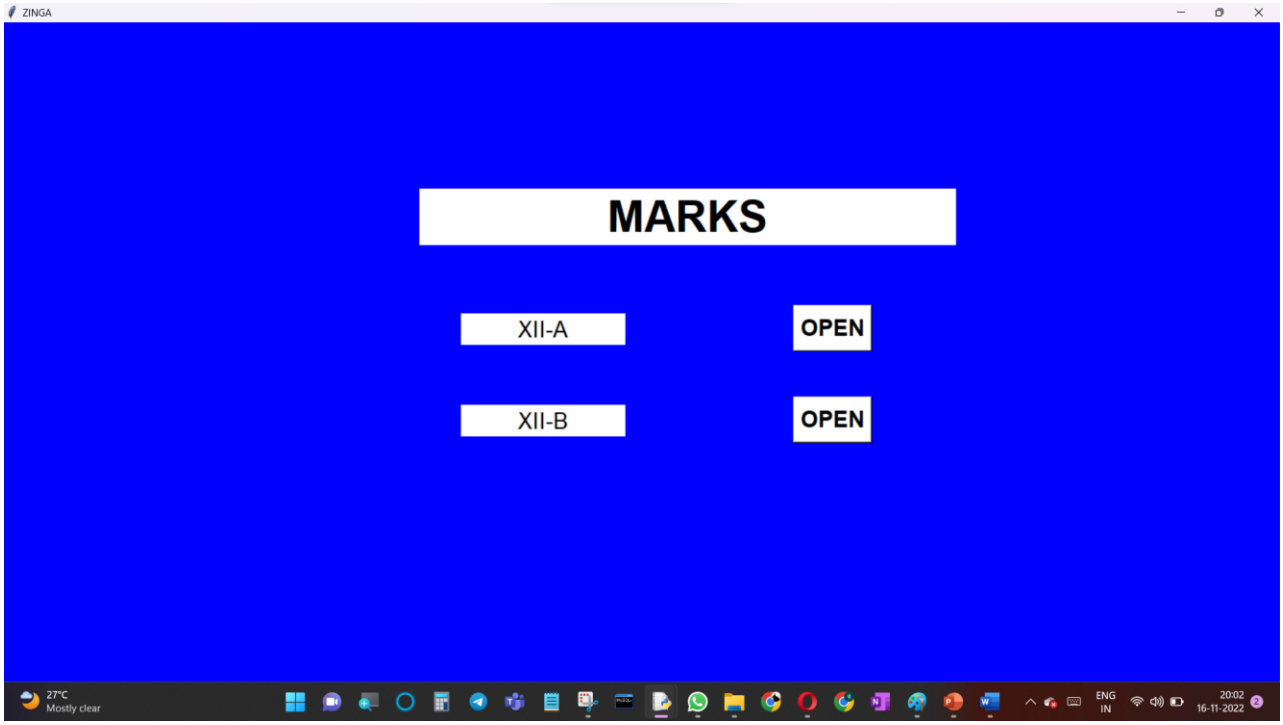
1) CREATING A LOGIN WINDOW



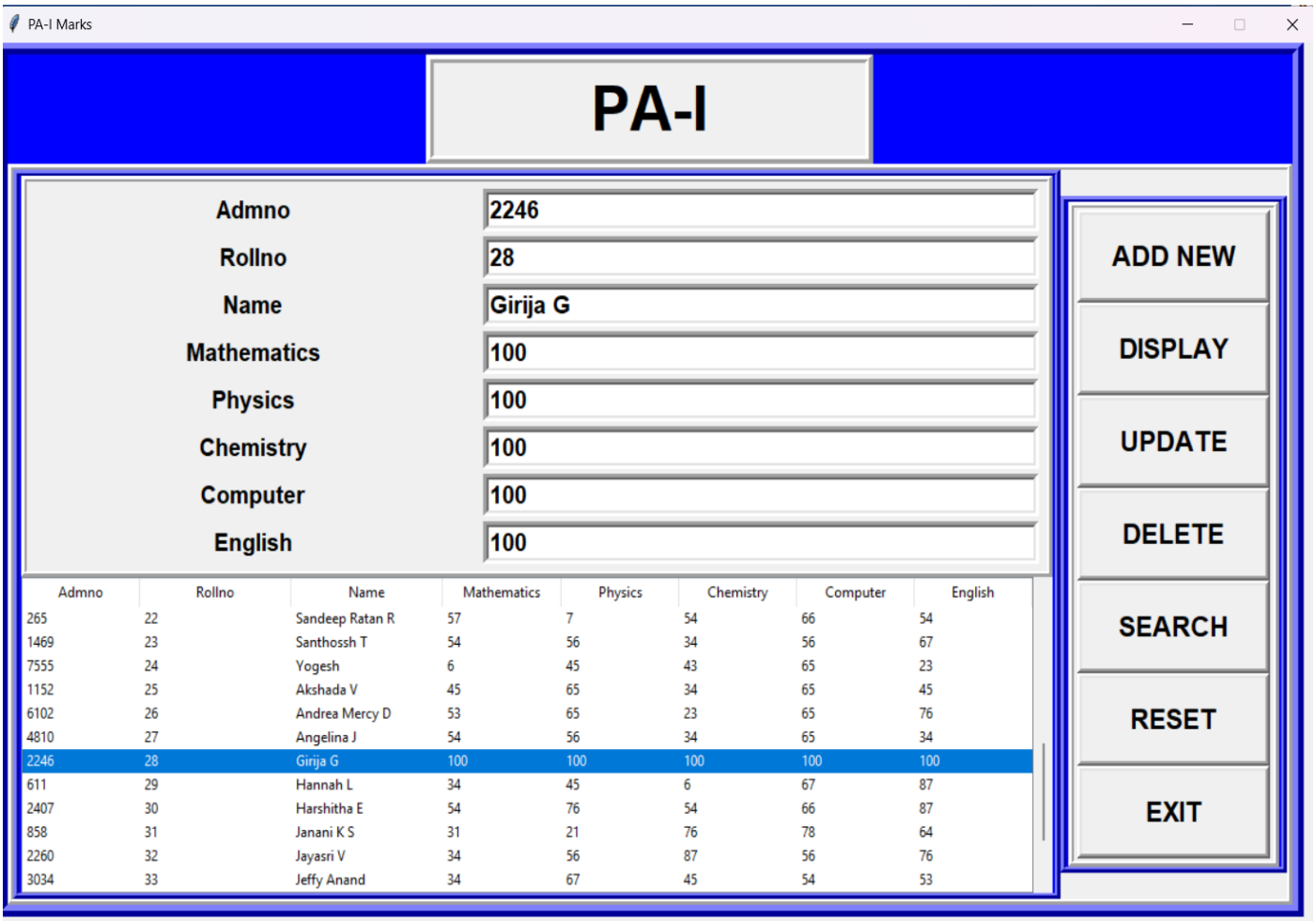
2) INCORRECT PASSWORD WINDOW



3) SECOND WINDOW



4)SQL WINDOW



Admno	Rollno	Name	Mathematics	Physics	Chemistry	Computer	English
265	22	Sandeep Ratan R	57	7	54	66	54
1469	23	Santhossh T	54	56	34	56	67
7555	24	Yogesh	6	45	43	65	23
1152	25	Akshada V	45	65	34	65	45
6102	26	Andrea Mercy D	53	65	23	65	76
4810	27	Angelina J	54	56	34	65	34
2246	28	Girija G	100	100	100	100	100
611	29	Hannah L	34	45	6	67	87
2407	30	Harshitha E	54	76	54	66	87
858	31	Janani K S	31	21	76	78	64
2260	32	Jayasri V	34	56	87	56	76
3034	33	Jeffy Anand	34	67	45	54	53

9. FUTURE ENHANCEMENT

In this application-based project we can add a few more features like giving a few many login IDs and passwords so that many more teachers or staff can use it, further we can few more classes and inside it. we can give options to create a new table so that scores for a few more tests can be accessed through the same applications side by side. And in order to move back to the previous window we can add another button in order to perform this task at the top.

10. BIBLIOGRAPHY AND REFERENCES

1) CREATING LABELS,BUTTONS,ENTRY WIDGET USING TKINTER
   https://www.youtube.com/watch?v=i5Iv8KU9rLU
2) https://www.youtube.com/watch?v=ScTgxrHqETI&t=936s
3) https://www.youtube.com/watch?v=nwht_3zUGI0
4) CREATING CONNECTIVITY WITH MYSQL FROM PYTHON
   https://www.youtube.com/watch?v=QbOFI1s0IXc
5) DISPLAYING CONTENT FROM SQL INTO TKINTER WINDOW
   https://www.youtube.com/watch?v=dxOPaIX4qt4
6) TO KNOW THE ARCHITECTURE OF TKINTER
   https://www.tutorialspoint.com/what-is-the-difference-between-the-widgets-of-tkinter-and-tkinter-ttk-in-python
7) grid( ) PARAMETERS
   https://www.tutorialspoint.com/python/tk_grid.htm#:~:text=ipadx%2C%20ipady%20−%20How%20many%20pixels,row%20that%20is%20still%20empty.
8) Messagebox in tkinter
   https://www.geeksforgeeks.org/python-tkinter-messagebox-widget/
9) Knowing more about tkinter
   https://docs.python.org/3/library/tkinter.html
10) knowing about __init__
    https://docs.python.org/3/library/tkinter.html
11) knowing about class and object
    https://www.geeksforgeeks.org/python-classes-and-objects/
12) syntax if __name__=’__main__’:

https://www.bogotobogo.com/python/python_if__name__equals__main__.php

13) Frames using tkinter

https://www.geeksforgeeks.org/python-tkinter-frame-widget/

14) StringVar()

https://www.oreilly.com/library/view/python-gui-programming/9781785283758/ch04s02.html

15) SUMITA ARORA WITH PYTHON CLASS 12